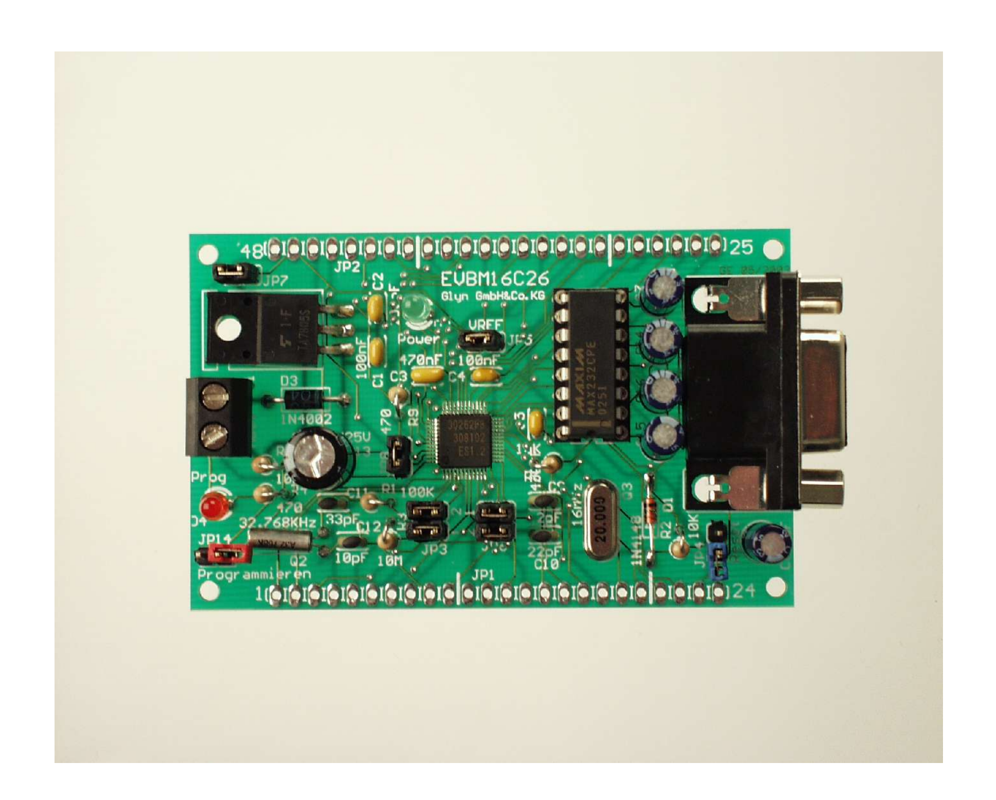
EVBM16C/26A

MANUAL

History

22th June 05	EW15	V1.0	Started
1st December 05	AP125	V1.1	New Version





Inhaltsverzeichnis

1.0Einleitung

Das Glyn EVBM16C/26A Evaluation Board ist ein Mini-Emulator, mit dem man einfach und preiswert mit den Features des M16C Mikrocontrollers eigene Applikationen entwickeln kann. Diese Mikrocontrollerfamilie wird von Renesas (ehemals Mitsubishi Electric) seit 1996 gefertigt und ist seit 2005 auch als M16C/26A erhältlich.

Das Applikationsprogramm darf bis zu 56Kbytes groß sein und dürfte für die meisten Applikationen ausreichen.
Zu den mitgelieferten Entwicklungswerkzeugen gehören KD30 Source-Level Debugger, Renesas C-Compiler, sowie das Flash Programm (FlashSta). Auf der CD sind speziell für dieses Tool vorbereitetes Beispiele, die mit dem Renesas C-Compiler compilierbar sind.

1.1 Lieferumfang

Dem EVBM16C/26A Board liegt folgendes bei:
1. dieses Manual
2. eine CD mit allen Softwaretools
3. das Board **mit** geflashtem Monitor.
4. RS232 Kabel
5. zwei 24-polige Stiftleisten zum auflöten

1.2 Kurzanleitung

Um die Firmware schnell auszutesten geht man folgendermaßen vor:

1. Serielle Verbindung zum PC herstellen und Spannung (8-15V) anschliessen.
2. KD30 starten
3. unter CDROM:\Sample_Code\ first.x30 laden.
4. GO im KD30 drücken.
5. Das Programm wird ausgeführt.

Wichtige Links:

MCU Datenblätter	http://www.renesas.com/eng/products/mpumcu/index.html
M16C Forum	http://www.m16c-forum.com/
Aktuelle M16C und Board Infos	http://www.m16c.de
MCU Tools	http://www.renesas.com/eng/products/mpumcu/tools/index.html
Renesas allgemein	http://www.eu.renesas.com
RTOS	http://www.segger.com
TCP/IP-Stack	http://www.segger.com
B/W, Color Display Treiber	http://www.segger.com
Mikrocontroller Forum / Beispiele	http://mikrocontroller.cco-ev.de/de/m16c.php
EMIT TCP/IP Software	http://www.emware.com
IAR C-Compiler	http://www.iar.com
Programmbeispiele	http://www.m16canz.com/sampcode.htm

2.0Hardware

2.1 technische Daten

Das Board wird mit 20MHz getaktet und verfügt über einen 5V Spannungsregler. Der nötige RS232 Pegelwandler MAX232 befindet sich ebenfalls auf der Platine. Für den Flash-Programming Modus ist ein spezieller Jumper, JP14 vorhanden und wird mit einer roten Leuchtdiode angezeigt.

2.2 Spannungsversorgung

Das Board wird über die 2 polige Anreihklemme angeschlossen. Eine unstabilisierte Spannung von 8-15V ist ausreichend. Es wird ein Strom von bis zu 100mA benötigt. Das Board kann man auch direkt über die Pfostenfeldstifte speisen (dazu JP7 ziehen). Liegt Spannung an, so leuchtet die grüne Leuchtdiode.

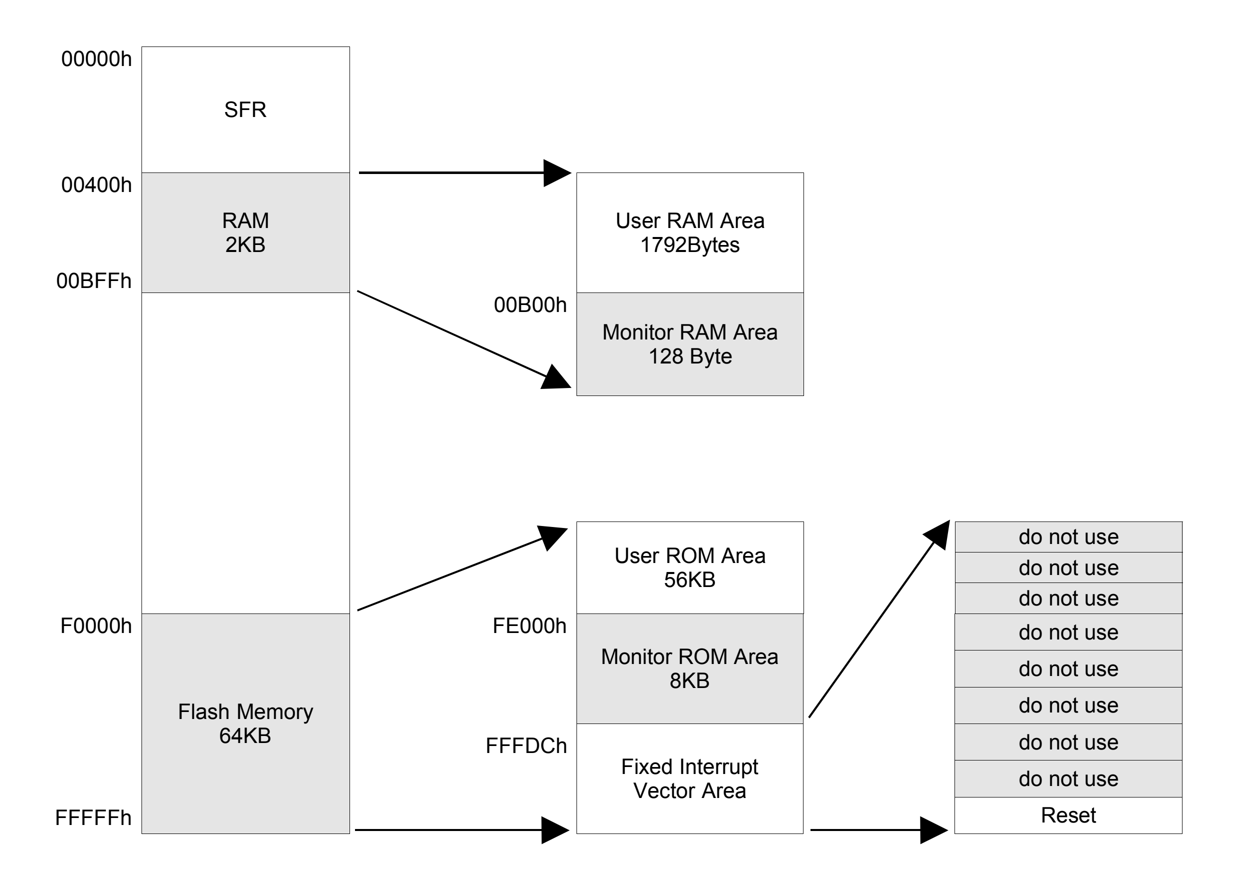
2.3 Jumper

JP8	Jumper für die Power LED. Damit kann für Strommessungen die LED ausgeschaltet werden.
JP1	Stiftleistenanschluß für die CPU-Pins 1-24
JP2	Stiftleistenanschluß für die CPU-Pins 25-48
JP3	Zwei Jumper für den SUB-Oszillator. Werden gesteckt, um die 32KHz Oszillatorschaltung auf dem Board zu verwenden. Wenn diese offen sind, werden P86 und P87 nur als Port-Pins verwendet.
JP5	Verbindet den VREF-Pin mit VCC. Bei Verwendung anderer Spannungen muß dieser offen sein.
JP7	Verbindet den Ausgang vom Spannungsregler mit der Schaltung. Sollte das Board über die Stiftleistenanschlüsse JP1/JP2 mit anderer Spannung gespeist werden (z.B. für eine Strommessung), so muß der Jumper offen sein.
JP4 / RESET	damit kann ein Reset ausgelöst werden.
JP14	Jumper zum programmieren. Im geschlossenen Zustand wird die MCU nach einem RESET in den Bootloadermode gebracht.
MAX232	der Chip ist gesockelt, um Strommessungen an der MCU zu ermöglichen (MAX232 Chip aus dem Sockel ziehen)

2.4 Anschlüsse

Auf dem Board befinden sich zwei Reihen mit je 24 Anschlüssen. **Jeder Anschluß entspricht dem jeweiligen Pin der MCU**. Dem Board sind zwei 24-polige Stiftleisten beigefügt, die man unter- oder auf das Board löten kann. Je nach Anwendungsfall.
Eine 9-polige D-SUB Buchse ermöglicht die Kommunikation mit der Umwelt.

2.5 Memory mapping



Für eigene Programme steht das SRAM von 0400h bis 0AFFh zur Verfügung. Das Flash ist für eigene Applikationen im Bereich von F0000h und FDDFFh nutzbar.

2.6 Anwenderhinweise

Bei jedem Download wird die MCU geflasht. Die Watchdog-, Adress Match-, NMI-, Overflow- und Undefined Instruction Interrupts sind bei dem Monitor nicht verwendbar.

3.0Software

3.1 mitgelieferte Software

NC30 C-Compiler V2.00............Ansi C-Compiler mit Assembler von Renesas. Ältere DOS-Version ohne Limitierung. Läuft nicht unter Win NT. Im Unterverzeichnis Sample_Code befinden sich fertig compilierte Demos für das EVBM16C/26 Board.

NC30 C-Compiler V5.10r1........ Neuste Version des Ansi C-Compilers von Renesas als Trial Version. Diese läuft 4 Monate. Bitte nicht in das NC30V2.00 Verzeichnis kopieren. Wird diese Version verwendet, müssen die Pfade zur V2.00 in der Autoexec.bat gelöscht werden. Zusätzlich die Vollversion des Toolmanagers für Windows.

KD30_Debugger .... Windows 98/2000/NT4 Source Level Debugger für Renesas- und IAR C-Compiler und Assembler.

EMI, EMV................................. EMI und EMV Design-Hinweise für Mikrocontroller

Datasheets................................ Datenblätter für den M16C/26A Mikrocontroller inkl. Software Manuals.

Flash_Start............................ Flash Programm von Renesas für das serielle *asynchrone* Flashen der Mikrocontroller. Ist notwendig um bei 'leeren', neuen MCUs den ROM-Monitor zu flashen, oder um später das eigene Programm ohne ROM-Monitor zu flashen.

Flasher...................................... Batchfähiges Flash Programm, ebenfalls für das serielle *asynchrone* Flashen der Mikrocontroller.

Application Notes...................... Hier sind 2 Application Notes: Flash programming und I²C Schnittstelle beschrieben.

Packages.................................. Gehäusemaßzeichnungen

Monitor...................................... ROM Monitor mit Source, fertig mit dem AS30 (Renesas) assembliert für den M30260F8AGP

Sample_Code...........................Programmbeispiele für die Peripherie der MCUs. Für den NC30 C-Compiler von Renesas vorbereitet.

Editor.. Editor für den Toolmanager TM30. Lässt sich mit %F -%L für Files und Zeilen einstellen.

3.2 Beschreibung der Monitorsoftware

Der Monitor belegt im Flash- Speicher den Bereich von FE000h bis FFFFFh und im SRAM den Bereich 0B00h bis 0BFFh. Der Monitor ist als Source und als Motorola-File auf der CD im Verzeichnis 'monitor' vorhanden. Die Monitorsoftware erwartet an UART1 die PC Debuggersoftware. Die Kommunikationsgeschwindigkeit beträgt 38400Bd. Der Monitor ist auf eine Quarzfrequenz von 20MHz eingerichtet, kann aber geändert werden. (siehe unter 3.1 Monitor)

3.3 Beschreibung der Flashersoftware Flashsta (Flasher_Mitsu)

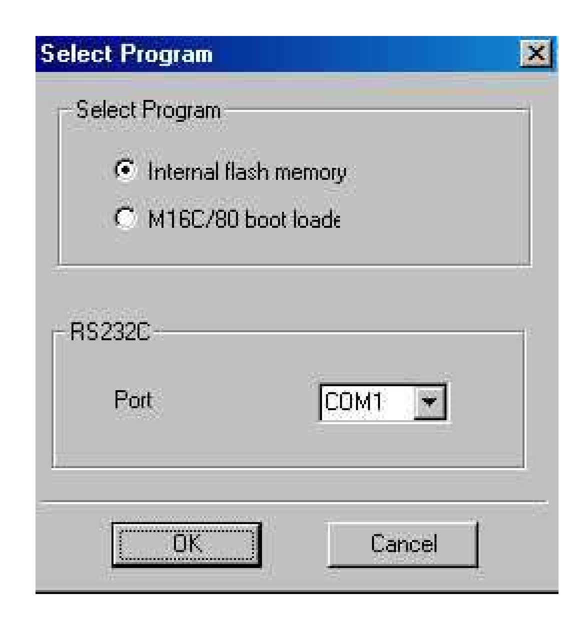
Vorab Info: Das Flash-Programm wird **erst am Schluß** der Entwicklung benötigt, der Monitor wurde schon auf das Board geflasht.
Man flasht damit das Applikationsprogramm oder einen neuen ROM-Monitor in die MCU. Deswegen kann man diesen Teil des Manuals erst mal überspringen.

Jeder Flash Mikrocontroller aus der M16C Serie verfügt über einen integrierten Bootloader. Bei dem M16C/26 ist dieser in einem zusätzlichen Flash-Speicher von 4Kbytes abgelegt.
Durch das schliessen des JP14 (programmieren) startet die MCU nach dem Reset diesen Bootloader.

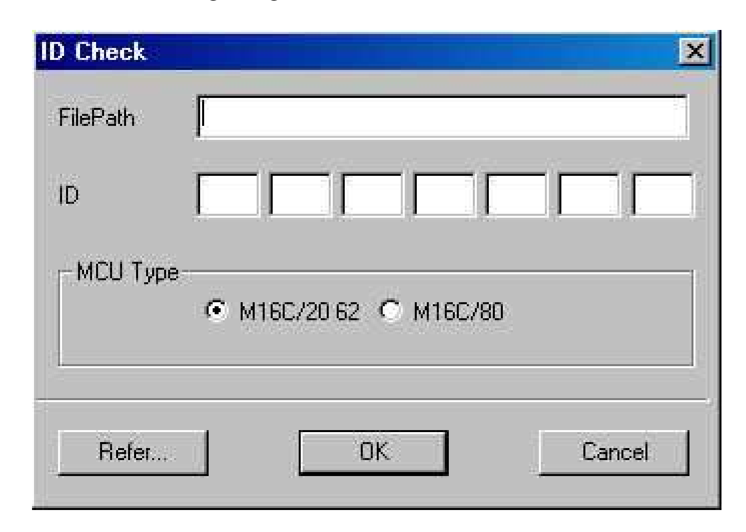
Ist der CLK1-Pin High, so geht der Bootloader in den seriellen synchronen Betrieb, ist er Low (wie in dieser Schaltung) geht er in den asynchronen Betrieb. Im Gegensatz zum synchronen Modus sind im asynchronen Modus sind nicht alle Quarzfrequenzen möglich, deshalb gibt Renesas im Datenblatt die Frequenzen an, mit denen es funktioniert.
Hier die möglichen Quarztakte und Baud-Raten für den M30260F8AGP:

20MHz	9600	19200	38400	57600
16MHz	9600	19200	38400	57600
12MHz	9600	19200	38000	---
11MHz	9600	19200	38000	---
10MHz	9600	19200	---	57600
8MHz	9600	19200	---	57600
7,3728MHz	9600	19200	38400	57600
6MHz	9600	19200	38400	---
5MHz	9600	19200	---	---
4,5MHz	9600	19200	---	57600
4,194304MHz	9600	19200	38400	---
4MHz	9600	19200	---	---
3,58MHz	9600	19200	38400	57600
3MHz	9600	19200	38400	---
2MHz	9600	---	---	---

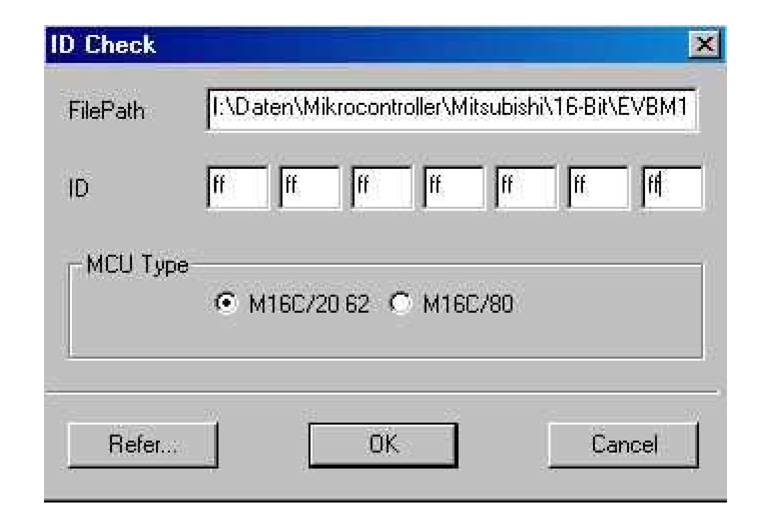
Flashersoftware starten:
Durch einen Doppelklick wird das Programm Flashsta.exe gestartet. Die detailierte Bedienungsanleitung zu dem Programm (in englisch) findet man im File manual_e.pdf.
Das Programm fragt jetzt nach der RS232 Schnittstelle des PCs an dem der Mikrocontroller angeschlossen ist. Bitte noch Internal Flash Memory auswählen und bestätigen.



Sollte es nach kurzer Zeit nach 9600 Bd fragen, so ist keine Verbindung zustande gekommen. Sind alle Verbindungen o.k. und greift kein zweites Programm auf die COM- Schnittstelle zu, so ist die Schaltung funktionsuntüchtig.
In diesem Fall sollte man noch einmal Reset drücken, den Oszillator-, Betriebsspannung- und am MAX232 die Spannungspegel prüfen. Eigentlich sollte nichts schief gehen. Falls man aber ein Fenster sieht, in dem nach einem Filenamen gefragt wird, hat alles funktioniert:



Bitte den Filenamen auswählen und bestätigen. Es **muss** ein Motorola Hex-File sein.
Das Flash-Programm Flashsta.exe wird sich jetzt über das Fehlen des ID-Files beschweren, falls das ID File nicht vom C-Compiler oder Assembler generiert wurde. Man kann es zunächst erst einmal wegklicken.
Eine Zeile tiefer sind 7 Felder zu sehen. Dort wir der Identifier des Mikrocontrollers eingegeben, der jetzt gerade läuft. Wurde er noch nie programmiert, oder ist er vorher gelöscht worden, gibt man 7 x FFh ein. Hat man ein Assemblerprogramm (z.B. unseren Monitor) oder ein C-Programm vorher geflasht, so gibt man 7 x 00h ein. Wurde vorher ein spezieller ID geflasht, so gibt man diesen ein. Hat man vorher ein Programm geflasht, so muss man in der Regel 7 x FFh eingeben.



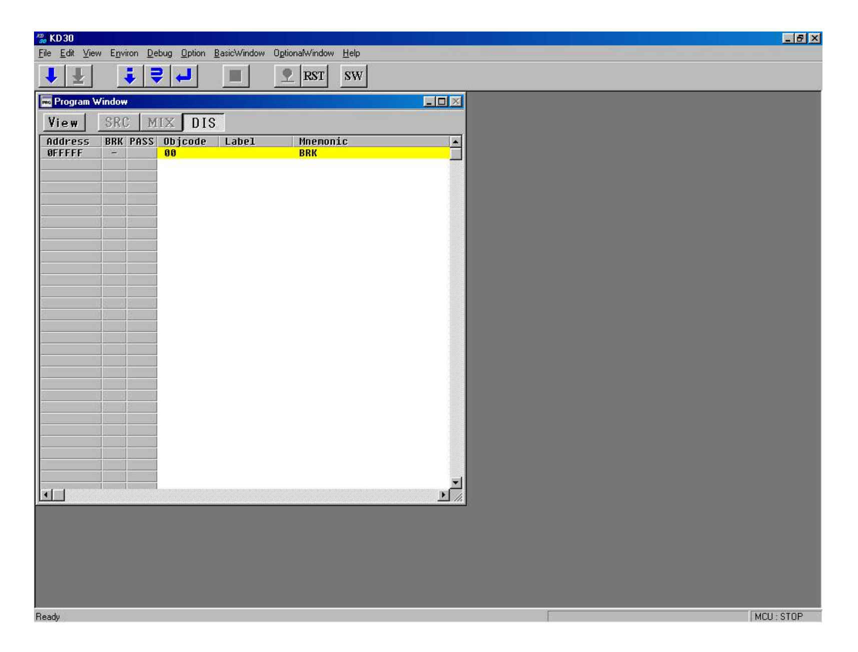
Ist alles in Ordnung., erscheint dieses Fenster. Falls nicht, war die ID falsch.



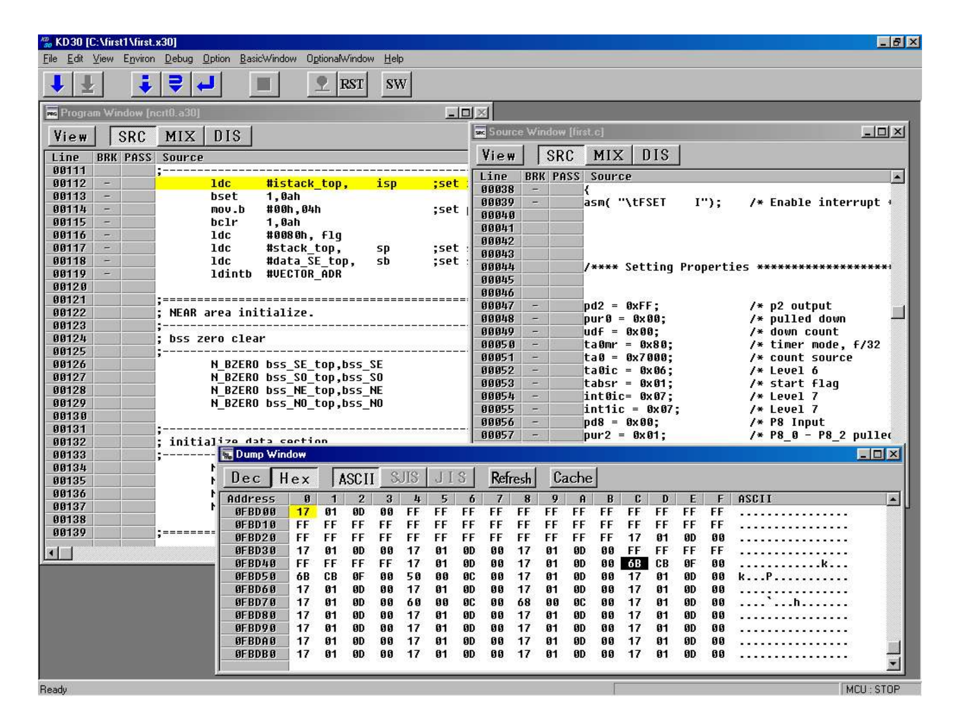
Mit 'Setting' stellt man nun die maximale Baud-Rate ein, die mit der Quarzfrequenz möglich ist.
Bei 16MHz sind das 57600Bd. Geht eine Baudrate nicht, so gibt es einen 'Fall-back' zu letzten funktionstüchtigen Baudrate. War der Chip nicht leer, so muß er jetzt mit Erase gelöscht werden.
Um das neue Programm (in diesem Fall den ROM-Monitor) zu flashen, benutzt man am besten 'E.P.R.' = Erase, Program, Read (verify). Sollte es zu keiner Fehlermeldung kommen, so ist der Chip ordnungsgemäß geflasht.
Nachdem das Programm beendet wurde, kann man jetzt die Betriebsspannung ausschalten, oder den Reset-Taster drücken und dabei dem Jumper ziehen.
Wird jetzt die MCU gestartet, so startet das User-Programm im User-Flash.

3.4 Beschreibung des Debuggers

Mit einem Doppelklick auf Setup.exe wird der Debugger installiert. Nach der Installation wird die COM Schnittstelle des PCs ausgewählt und mit OK bestätigt (das EVBM16C/26A muss angeschlossen und eingeschaltet sein). Die detaillierte **Bedienungsanleitung** zu dem Programm (in englisch) findet man im Help-File. Das Ergebnis müsste dann so aussehen:



Über das Pull-down Menü 'File → download → Load Module...' kann man jetzt das mitgelieferte und compilierte Programm CDROM:\Sample_Code\first\first.x30 (vom NC30) downloaden und anzeigen lassen. Das Program-Window (links) folgt immer dem PC, das Source Window (wird über BasicWindow → Source Window geöffnet) zeigt immer die beobachtete Funktion an. In diesem Fall wird *main* angezeigt.
Unten sieht man das Dump Window (wird über BasicWindow → Dump Window geöffnet) ab der Adresse FDD00, der Anfangsadresse der variablen Interrupt Vektoren.



Auf der Benutzeroberfläche befinden sich Buttons, wie
"GO" (Programmstart ab Program Counter),
"Come" (Go bis cursor),
"Step" (Einzelschritt),
"Step over Subroutine",
"Step from Subroutine",
"Stop" (hält laufendes Programm an),
"RST" (Reset des Systems),
"BREAK-points" (listet alle Breakpoints auf)
um nur die wichtigsten zu nennen.
Zum debuggen ist es möglich, neben dem Program-Window auch mehrere Source-Windows zu öffnen. In dem Script-Window können Scripte eingegeben und abgespeichert werden. Damit kann z.B. eine persönliche Arbeitsumgebung für den Start des Emulators initialisiert werden. Der Emulator frischt einmal pro Sekunde den Bildschirm auf. D.h. RAM Monitor-Window und Watch-Window werden ständig aufgefrischt. Damit das die laufende Applikation nicht stört, kann man die refresh-Zeit erhöhen, oder diese Funktion abschalten (**Free-Run Mode**).

Eine detailierte Beschreibung befindet sich nach der Installation
unter 'Start' → 'Programme' → 'RENESAS-TOOLS' → 'KD30 V4.10 Release 1' → 'KD30 Users Manual.pdf'

3.5 Der Renesas C-Compiler V2.0 (DOS-Version)

Renesas C-Compiler: Um den C-Compiler zu installieren geht man wie folgt vor:
1. Verzeichnis c:\mtool erzeugen
2. nc30.zip dorthin kopieren
3. mit pkunzip -d nc30 oder winzip dort entpacken
4. die Textzeilen von autoexec.mit ins eigene autoexec.bat kopieren

Kurzanleitung zum Laden des 'first' Projekts:
Auf der CD im Verzeichnis NC30 C-Compiler V2.00/ first das File first.zip entpacken und auf der Festplatte z.B. unter c:\mtool\first installieren.
Dann im DOS-Modus 'comp' eingeben. Nun wird das Programm first.c wird compiliert.
Die Anpassungen für den ROM-Monitor sind in den Files sect30.inc (Vektoren, Mapping) und nrt0.a30 (Startup Assemblerfile) schon enthalten.

Hier die Vorgehensweise:

Anpassen der Startup Files an den ROM-Monitor:

ncrt0.a30:

```
;------------------------------------------------------------------------------------------------
; INTERRUPT VECTOR ADDRESS  definition                    von 0FFD00h in 0FDD00h ändern
;------------------------------------------------------------------------------------------------
VECTOR_ADR      .equ      0fdd00h            ;special vector adress for monitor

start:
;------------------------------------------------------------------------------------------------
; after reset,this program will start                     hier mov.b #08h,05h eintragen
;------------------------------------------------------------------------------------------------
          ldc       #istack_top,isp          ;set istack pointer
          bset      1,0ah
          mov.b     #00h,04h                          ;set processor mode
          bclr      1,0ah
          bset      0,0ah
          mov.b     #20h,07h                          ;set clock control register
          mov.b     #08h,06h                          ;to no clock division
          bclr      0,0ah
          ldc       #0080h,   flg
          ldc       #stack_top,sp            ;set stack pointer
          ldc       #data_SE_top,sb          ;set sb register
          ldintb    #VECTOR_ADR
```

sect30.inc:

```
;------------------------------------------------------------------------------------------------
; Far ROM data area                                              hier  0E0000h eintragen
;------------------------------------------------------------------------------------------------
                .section  rom_FE,ROMDATA
                .org      0f0000h;

diese Zeilen  eintragen:
;------------------------------------------------------------------------------------------------
; variable vector section                                 von dummy_int in 0FF900h ändern
;------------------------------------------------------------------------------------------------
                .lword    0FF900h                     ; uart1 trance (for user)
                .lword    0FF900h                     ; uart1 receive (for user)      .
```

Erklärung:
Die Vektoradresse wird ebenfalls wie beim Tasking C-Compiler auf 0xFDD00 eingetragen. In diesem Beispiel wird hier schon einmal das Clock-Register auf /1 gesetzt.
Am Schluß wird noch der UART1 Vektor 0FF900h eingetragen, damit der ROM-Monitor nach dem laden des USER-Programms seine Verbindung über UART1 halten kann.

Aufruf des C-Compilers in folgender Syntax

Command Line: nc30 -silent -v **-g** -o*outputname* -lnc30lib.lib ncrt0.a30 *filename*.c

Input: *filename*.c : das auszuführende C-Programm
lnc30lib.lib : Standard Libary
ncrt0.a30 : startup program incl. sect30.inc (Initialisierung des heap, stack, ...)

Output: *filename*.x30 : machine language file des erstellten C-Programmes
filename.map : map file mit Informationen für den Debugger

Parameter: -silent : unterdrückt die Copyright-Meldung
-v : teilt den Namen des ausführenden Programmes mit
-g : erzeugt Informationen für den Debugger
-o*outputname* : gibt den Namen des zu erzeugenden X30 files an
-llnc30.lib : bindet Standard Libary file für den Linker ein

Um zum flashen ein Hex-File zu erzeugen, kann mit dem Aufruf 'lmc30 *filename'* ein Motorola HEX-File erzeugt werden.

Zum debuggen ist es erforderlich, dass das Interrupt Flag gesetzt ist. Damit man im C-Programm debuggen kann, muss dieses dort auch gesetzt werden.

Beispiel:

```
void far main()
{
timerinit ();
asm( "\tFSET   I");       /* Enable interrupt */
.
.
.
.
}
```

Aber Vorsicht, man sollte das erst nach der Initialisierung eigener Interrupts eintragen.
(Siehe Seite 15 bzw.16 unter Punkt 10)

Noch ein Hinweis:
Man kann mit dem Befehl printf........ Daten seriell oder parallel übertragen. Bei dem NC30 C-Compiler ist das sogar schon voreingestellt. Leider verwendet Renesas als Standard Ausgabe die UART1, also die serielle zum debuggen. Man kann das in der Library ändern oder, um es abzuschalten die folgenden Zeilen im NCRT0.A30 auskommentieren:

```
;====================================================================
; Initialize standard I/O
;---------------------------------------------------------------------
;          .glb     _init
;          jsr.a    _init
```

3.6 Der Renesas C-Compiler V5.10

Die Weiterentwicklung des Renesas Compilers kann aus dem Internet geladen werden und hat zur Zeit die V. 5.10r1. (http://www.renesas.com/eng/products/mpumcu/toolhp/trial/nc30ty_e.htm)
Um diesen zu installieren geht man wie folgt vor:
(die NC30 Version 2.00 darf zwar dann noch in einem anderen Verzeichnis auf dem Rechner sein, die Pfade in der Autoexec.bat müssen aber vorher ausgetragen werden)

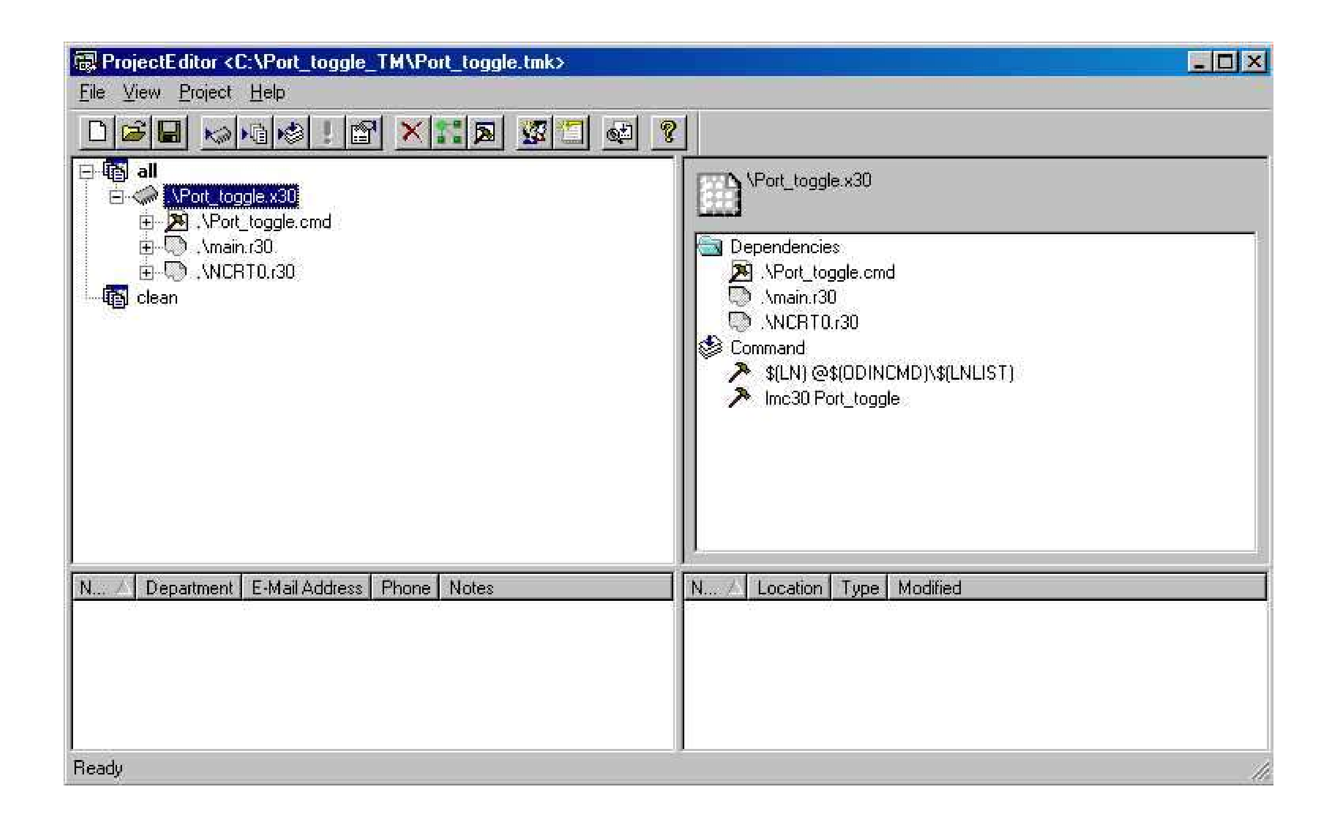
1. Doppelklick auf das File nc30wav500r1_e (Installiert die Windows-Version), und dann Trial-Version auswählen.
2. Doppelklick auf das File tm320a_e (Installiert den Toolmanager)

Kurzanleitung zum Laden des 'first' Projekts (läuft mit NC30V2.00 und NC30V5.10):
Auf der CD im Verzeichnis sample-program das File first.zip entpacken und auf der Festplatte installieren.
Dann im DOS-Modus 'comp' eingeben. Nun wird das Programm first compiliert.
Wichtige Anpassungen an den ROM-Monitor sind in den Files sect30.inc (Vektoren, Mapping) und nrt0.a30 (Startup Assemblerfile) schon enthalten.

Mit dem C-Compiler NC30V510 wird auch der TM3.20a ausgeliefert. Dies ist eine Benutzeroberfläche, die für Windows Rechner eingesetzt werden kann.
Ist der Toolmanager installiert und gestartet, so meldet er sich mit dieser Leiste:



Wird mit dem Button das Projekt Port_toggle von der CD geöffnet, so wird der Projekteditor gestartet:



Alle weiteren Informationen zu diesem Programm kann man unter Start --> Programme --> RENESAS-TOOLS --> TM V.3.20a --> TM User's Manual.pdf entnehmen.

4.0 Tipps und Tricks

Wie bekommt man später das Programm ohne Monitor zum laufen ?

Zunächst muß der UART1 Vektor im File sect30.inc auf dummy_int gesetzt werden. (Ist nicht unbedingt erforderlich, sollte aus Sicherheitsgründen aber gemacht werden.
Als zweites stellt man im File ncrt0.a30, einige Zeilen nach dem Label 'start: ' den Clockgenerator von / 8 auf /1 (Protection aufheben nicht vergessen), damit Ihre Applikation nicht plötzlich mit 1/8 der Geschwindigkeit läuft.
Und als drittes fügen Sie dem Batch-File comp.bat die Zeile ' lmc30 first ' ein (das erzeugte x30-File). Diese erzeugt dann ein Motorola Hex-File, welches Sie zum flashen mit dem oben genannten Flash-Programm benötigen.

Weitere Tipps gibt es unter http://www.m16c.de

Hinweise zur Applikationserstellung mit dem M16C/26A

Um den M16C in der Schaltung ordnungsgemäß zum arbeiten zu bringen sind einige wichtige Dinge zu beachten:

1. Der Anschluss AVCC ist intern etwas hochohmiger mit dem Chip verbunden sind und muss außen mit dem VCC verbunden werden.
   Ein Abblockkondensator ist aber dringend erforderlich. Der Anschluss ist direkt mit dem A/D-Wandler verbunden.
2. Offenliegende Portleitungen müssen, um Strom zu sparen entweder extern einen Pullup/down oder intern einen Pullup/down (programmiert) bekommen, oder der offene Port muß auf Ausgang geschaltet werden.
3. Die A/D-Wandler Eingänge sind normalerweise sehr hochohmig (im Megaohmbereich), während dem Messvorgang sind es aber für einige Nanosekunden nur ca 7,8 KOhm. Deshalb ist bei jedem A/D-Wandlereingang ein Kondensator (ca 100pF) vorzusehen.
4. Um einen 32KHz Quarz extern zu nutzen, sind die Portleitungen P86 und P87 mit Software auf die XCIN und XCOUT umzuschalten. Dazu muß aber das Clock-Register über das Protect-Register freigeschaltet werden.
5. Die CPU ist nach dem RESET auf Takt/8 geschaltet. Das kann man im Clock-Register auf Takt/1 ändern. (Protectregister beachten)
6. Interrupts (Priorität, ein und aus) dürfen innerhalb eines Interrupts nur dann verändert werden, wenn die Programmfolge
   ```
   FCLR I
   AND.B #00h, 055h (OR, BSET, BCLR geht auch)
   NOP
   NOP
   FSET I
   ```
   eingehalten wird. MOV ist verboten! Der Grund ist die instruction queue, die erst geleert werden muß.
7. Soll der Software-Reset ausgelöst werden (Flag im Prozessor-Mode-Register 0), muß vorher der Clock auf /8 geschaltet werden.
8. Falls für die serielle Datenübertragung kein MAX232 sondern eine andere Schaltung verwendet wird, muß überprüft werden, ob diese ebenfalls über einen Pull-up Widerstand am TTL-Eingang verfügt. Falls nicht, muß ein Widerstand (ca 470kOhm) zwischen TXD1 und VCC geschaltet werden. Grund: Der Monitor, sowie der Bootloader lösen im Betrieb öfter einen Software-RESET aus. Da der TXD1 Pin dann für Sekundenbruchteile Input wird und floatet, kann es zu fehlerhaften Datenübertragungen kommen.

EVBM16C/26A Bauteilliste:

Pos.	Bauteil	Typ	Wert	Raster (mm)	Reichelt Bezeichnung*
1	C1	100nF	Vielschicht-Kondensator	2,5	Z5U-2,5 100n
2	C2	100nF	Vielschicht-Kondensator	2,5	Z5U-2,5 100n
3	C3	470nF	Vielschicht-Kondensator	2,5	Z5U-2,5 470n
4	C4	100nF	Vielschicht-Kondensator	2,5	Z5U-2,5 100n
5	C5	470nF	Vielschicht-Kondensator	2,5	Z5U-2,5 470n
6	C6	1µF/6,3V	Elko / Tantal	2,5	rad 1/35
7	C7	4,7µF/16V	Elko / Tantal	2,5	rad 4,7/35
8	C9	22pF	Keramik-Kondensator	2,5	NPO-G1206 22P
9	C10	22pF	Keramik-Kondensator	2,5	NPO-G1206 22P
10	C11	33pF	Keramik-Kondensator	2,5	NPO-G1206 33P
11	C12	10pF	Keramik-Kondensator	2,5	NPO-G1206 10P
12	C13	220µF/25V	Elko	5	rad 220/25
13	C14	4,7µF/16V	Elko / Tantal	2,5	rad 4,7/35
14	C15	4,7µF/16V	Elko / Tantal	2,5	rad 4,7/35
15	C16	4,7µF/16V	Elko / Tantal	2,5	rad 4,7/35
16	D1	1N4148	Diode	10	SMD 1N4148
17	D2	LED 3mm grün	Leuchtdiode	2,5	LED 3mm gn
18	D3	1N4002	Diode	10	1N4002
19	D4	LED 3mm rot	Leuchtdiode	2,5	LED 3mm rt
20	G1	2 pol. Anreihklemme	Anreihklemme	5	ALK 101-2
21	IC1	MAX232 DIP	MAX232P	DIP	MAX232
22	IC2	7805T	Spannungsregler TO220	2,54	Glyn TA7805S
23	JP1	Stiftleiste 1x24	Stiftleiste	2,54	Stiftl. 36G
24	JP2	Stiftleiste 1x24	Stiftleiste	2,54	Stiftl. 36G
25	JP3	Stiftleiste 2x2 + Jumper	Jumper (Sub-Oszillator)	2,54	2 x Jumper schw.
26	JP4	Stiftleiste 1x2 + Jumper	Jumper (Reset)	2,54	Jumper schw.
27	JP5	Stiftleiste 1x2 + Jumper	Jumper (VREF)	2,54	Jumper schw
28	JP7	Stiftleiste 1x2 + Jumper	Jumper (VCC)	2,54	Jumper schw.
29	JP8	Stiftleiste 1x2 + Jumper	Jumper (Power-LED)	2,54	Jumper schw.
30	JP14	Stiftleiste 1x2 + Jumper	Jumper (Programmieren)	2,54	Jumper schw.
31	Q2	Quarz 32,768KHz	Uhrenquarz	2,5	Glyn VT-200-F
32	Q3	Quarz 20MHz	HC49/S	5	20-HC49U-S
33	R1	100 KΩ	Widerstand 1/4W	10	1/4W 100K
34	R2	10 KΩ	Widerstand 1/4W	10	1/4W 10K
35	R3	10 MΩ	Widerstand 1/4W	10	1/4W 10M
36	R4	470 Ω	Widerstand 1/4W	10	1/4W 470
37	R5	10 KΩ	Widerstand 1/4W	10	1/4W 10K
38	R6	10 KΩ	Widerstand 1/4W	10	1/4W 10K
39	R7	1 MΩ	Widerstand 1/4W	10	1/4W 1M
40	R9	470 Ω	Widerstand 1/4W	10	1/4W 470
41	U$1	M30260F8AGP	MCU	0,5	Glyn M30260F8AGP
42	X1	F09H	D-SUB Buchse 9-Pol	2,5	Glyn DRB-09S
43	---	Stiftleiste für Jumper		2,54	Stiftl. 36G
44	---	Sockel 16-pol.	Sockel für MAX232	2,54	GS16P

* Reichelt Elektronik
  Elektronikring 1
  26452 Sande
  Tel.: 04422-955-333 Fax.: 04422-955-111



Am Wörtzgarten 8
65510 Idstein/Ts.
- Germany -
Tel. 06126-590-222
Fax 06126-590-111


GLYN
Am Würtzgarten 8
65510 Idstein/Ts.
- Germany -
Tel. 06126-590-222
Fax 06126-590-111